UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/645,699	08/20/2003	Yutaka Enko	16869S-091100US	9300

20350   7590   05/01/2008
TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER
MYINT, DENNIS Y

ART UNIT	PAPER NUMBER
2162	

MAIL DATE	DELIVERY MODE
05/01/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on _28 February 2008_.
2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under _Ex parte Quayle_, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) _1,2,4,5,9,12 and 14-24_ is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) _1, 2, 4, 5, 9, 12, 14-20, and 21-24_ is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.
10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a) ☐ All b) ☐ Some * c) ☐ None of:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____ .

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1.	A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on February 28, 2008, has been entered.

2.	The amendment filed on February 28, 2008 has been received and entered. Claims 1, 2, 4, 5, 9, 12, 14-20, and 21-24 are currently pending in this application. In the amendment filed on February 28, 2008, claims 1, 4, 9, and 12 were amended and claims 21-24 were newly added. Claims 1, 9, 12, and 21 are independent claims.

Response to Arguments

3.	The applicant's arguments filed on February 28, 2008 have been fully considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 112

4.	The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5.   Claims 1, 2, 4, 5, 9, 12, 14-20, and 21-24 are rejected under 35 U.S.C. 112,

second paragraph, as being indefinite for failing to particularly point out and distinctly

claim the subject matter which applicant regards as the invention.

As per claim 1, the claim in lines 34-35 recites "when the first computer receives

an access request for the file after the file has been transferred to the second

computer". However, it is not ascertainable which computer made said access request

because the claimed system is comprised of at least three computers. Therefore, said

limitation renders the whole claim indefinite and claim 1 is rejected under 35 U.S.C. 112

second paragraph for failing to particularly point out and distinctly claim the subject

matter which applicant regards as the invention.

Claim 1 in lines 37-38 also recites "a module for receiving a return request packet

from the second computer". However, it is not ascertainable which computer receives

said return request packet because the claimed system is comprised of at least three

computers. Therefore, said limitation renders the whole claim indefinite and claim 1 is

rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and

distinctly claim the subject matter which applicant regards as the invention.

Claim 1 also recites in lines 37-38 "issuing a read request in response". However,

it is not ascertainable which computer issued said read request in response because

the claimed system is comprised of at least three computers. Therefore, said limitation

renders the whole claim indefinite and claim 1 is rejected under 35 U.S.C. 112 second

paragraph for failing to particularly point out and distinctly claim the subject matter which

applicant regards as the invention.

Claims 2, 4, 5, and 14-20 depend on claim 1 and are also rejected under 35 U.S.C. 112 second paragraph by virtue of their dependency on claim 1.

As per claim 9, the claim in lines 32-33 recites "when the first computer receives an access request for the file after the file has been transferred to the second computer". However, it is not ascertainable which computer made said access request because the claimed system is comprised of at least three computers. Therefore, said limitation renders the whole claim indefinite and claim 9 is rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 9 in lines 35-36 also recites "a module for receiving a return request packet from the second computer". However, it is not ascertainable which computer receives said return request packet because the claimed system is comprised of at least three computers. Therefore, said limitation renders the whole claim indefinite and claim 9 is rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 9 also recites in lines 35-36 "issuing a read request in response". However, it is not ascertainable which computer issued said read request in response because the claimed system is comprised of at least three computers. Therefore, said limitation renders the whole claim indefinite and claim 9 is rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claim 12, the claim in lines 28-29 recites "when the first computer receives an access request for the file after the file has been transferred to the second computer". However, it is not ascertainable which computer made said access request because the claimed system is comprised of at least three computers. Therefore, said limitation renders the whole claim indefinite and claim 12 is rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 12 in lines 31-32 also recites "a module for receiving a return request packet from the second computer". However, it is not ascertainable which computer receives said return request packet because the claimed system is comprised of at least three computers. Therefore, said limitation renders the whole claim indefinite and claim 12 is rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 12 also recites in lines 31-32 "issuing a read request in response". However, it is not ascertainable which computer issued said read request in response because the claimed system is comprised of at least three computers. Therefore, said limitation renders the whole claim indefinite and claim 12 is rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claim 21, the claim in lines 25-26 recites "when the first computer receives

an access request for the file after the file has been transferred to the second computer". However, it is not ascertainable which computer made said access request because the claimed system is comprised of at least three computers. Therefore, said limitation renders the whole claim indefinite and claim 21 is rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 21 in lines 27-28 also recites "receiving a return request packet from the second computer". However, it is not ascertainable which computer receives said return request packet because the claimed system is comprised of at least three computers. Therefore, said limitation renders the whole claim indefinite and claim 21 is rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 21 also recites in lines 27-28 "issuing a read request in response". However, it is not ascertainable which computer issued said read request in response because the claimed system is comprised of at least three computers. Therefore, said limitation renders the whole claim indefinite and claim 21 is rejected under 35 U.S.C. 112 second paragraph for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 22-24 depend on claim 21 and are also rejected under 35 U.S.C. 112 second paragraph by virtue of their dependency on claim 21.

Claim Rejections - 35 USC § 103

6.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or
> described as set forth in section 102 of this title, if the differences between the subject
> matter sought to be patented and the prior art are such that the subject matter as a whole
> would have been obvious at the time the invention was made to a person having ordinary
> skill in the art to which said subject matter pertains. Patentability shall not be negatived
> by the manner in which the invention was made.

7.   This application currently names joint inventors. In considering patentability of

the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of

the various claims was commonly owned at the time any inventions covered therein

were made absent any evidence to the contrary. Applicant is advised of the obligation

under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was

not commonly owned at the time a later invention was made in order for the examiner to

consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g)

prior art under 35 U.S.C. 103(a).

8.   Claims 1, 2, 4, 5, 9, 12, 14, 15, 16, 17, 19, 20, 21, 22, 23, and 24 are rejected

under 35 U.S.C. 103(a) as being unpatentable over Rabinovich (hereinafter

"Rabinovich, U.S. Patent Number 6256675) in view of Fung et al., (hereinafter "Fung",

U.S. Patent Application Publication Number 2007/0136393).

As per claim 1, Rabinovich is directed to a computer system comprising a first network, a first computer connected to the first network, a second network connected to the first network, and a second computer and a third computers connected to the second network"(Rabinovich Figure 1: *Host 103, Host 104,Host 105, and Request Distributor 101,* and Column 7 Lines 7-49. Note that Requestor Distributor 101 and a Host 103 of Figure 1 of Rabinovich maps to the first computer of the claimed invention; Host 104 and Host 105 of Figure 1 of Rabinovich maps to the second computer of the claimed invention and Requester 109 of Figure 1 of Rabinovich maps to the third computer of the claimed invention *),* the first computer (Request Distributor 101 and Host 103) comprises:

"a communication interface for connecting the first computer to the first network" (Rabinovich Figure 1: PORT 110 , and Column 6 Lines 31-33, i.e., *Request distributor 110 also includes a port 110 that is adapted to be coupled to a network 102*);

"a disk storage device for storing data" (Rabinovich Figure 1: *Request distributor 101,* and Column 6 Lines 17-29, i.e., *The request distributor is comprised of a processor 106, and a memory 107 that stores request distribution instructions;* Column 6 Lines 47-49, i.e., *embodiments of memory in accordance with the present invention include a hard disk drive*);

"a disk interface for communicating data with the disk storage device" (Rabinovich Figure 1: *Request Distributor 101,* Column 6 Lines 17-29 and Column 7 Lines 7-49);

"a CPU for controlling the first computer" (Rabinovich Figure 1: *Request distributor 101,* and Column 6 Lines 7-16; and Figure 1: *Host 103;* and Column 6 Line 60 through Column 7 Line 7) ; and

"a memory" (Rabinovich Column 6 Lines 17-29, i.e., *The request distributor is comprised of a processor 106, and a memory 107 that stores request distribution instructions*) "for storing data and a first program" (Rabinovich Column 6 Lines 60-62, i.e., *a request distribution method;* and Figure 2) and "a second program for operating the CPU" (Rabinovich Column 7 Line 38 through Column 8 Line 31, i.e., *The request distributor selects a host that stores a replica of the requested object to respond to the request based upon the request metric and the distance metric;* and *the request distribution decision as to which host to assign the request is made in accordance with the method shown in Figure 3*),

wherein the program code includes:

"a module for recording situations of access to a file stored in the disk storage device from the third computer" (Rabinovich Column 6 Lines 64-67, i.e., *The request metric for a replica is a historical measure of the request for the object that have been forwarded to the host*), "the module being executed by the CPU at predetermined intervals" (Rabinovich, Column 11 Lines 66-67, i.e., *load metrics are usually averaged over some sampling interval preceding the time the metric is recorded*) and

"wherein the program code is executed depending on the access situation" (Rabinovich Column 7 Lines 38-44, i.e., *The request distributor selects a host that*

stores a replica of the requested object to respond to the request based upon the

request metric and the distance metric), "the program code further including:"

"a module for searching the second network connected to the third computer"

(Rabinovich Column 7 Line 45 through Column 8 Line 31, i.e., *the request distributor*

decision as to which host to assign the request is made in accordance with the method

shown in FIG. 3);

"a module for searching candidate for migration for the second network"

(Rabinovich, Column 9 Lines 9-23, i.e., *a method for identifying a second host to which*

it is beneficial to migrate a replica from a first host to the second host is shown in Figure

8. Rabinovich, Column 7 Line 45 through Column 8 Line 31, i.e., *the request distributor*

decision as to which host to assign the request is made in accordance with the method

shown in FIG. 3);

"a module for designating the files as the candidate for migration to the second

computer" (Rabinovich Column 7 Lines 47-49, i.e., *A host p is identified that stores* ***a***

replica *of the requested object and that has the best distance metric m;* Column 8 Lines

29-31, i.e., *The request for the object is sent to the host with the smallest decision*

metric, step 403; Column 16 Lines 19-22, i.e., *when object X3 is to be replicated or*

migrated, S attempts t place the replica on the farthest among all qualified candidates.

Column 17 Lines 28-31, i.e., *when host S is in Offloading mode, it migrates or replicates*

objects to other nodes. Note that by identifying host *p* as the candidate, the replica on

host p is also identified as the candidate for migration);

"a module for transmitting a migrator acceptor search packet to the second

computer for inquiring whether or not the second computer can accept the file"

(Rabinovich Column 7 Line 64, i.e., *then the request is sent to host p;* and Column 8

Lines 32-49, i.e., *Replica placement decisions and actions are made and taken*

substantially autonomously by a host; Column 8 Line 65 through Column 9 Line 8, i.e.,

a method for offloading in accordance with the present invention is shown in FIG 7;

Column 16 Lines 23 through Column 17 Line 22, i.e., *Host s sends a replication request*

to E's replicator r_E , which includes the ID of the object to be replicated and the load on

host s generated due to x_3. r_E forwards this request along the path with the lowest-

loaded hosts and pseudo code from Line 28 to 54; Particularly Column 16 Lines 29-31,

i.e., *ReplicateRequest (x_s, load (x_3));*   );

"a module for receiving a reply packet from the second computer as a response

to the migrator acceptor search packet" (Rabinovich, Column 16 Lines 56-65, i.e.,

sends acknowledgement back up the tree to r_E and Column 16 Lines 30-35, i.e.,

Send Refuse to invoker; Column 18 Lines 19-20, i.e., *Upon receiving Recipient request,*

host q starts the second stage of the protocol by sending OffloadOffer to s);

"a module for transmitting advertisement packets to the third computer either

after **or** before the file is transferred to the second computer, the advertisement packet

indicating that the file is transferred to the second computer (Rabinovich Column 15

Line 17, i.e. **send** *OffLoadRequest(s) to the parent* **replicator** *of s;* and Column 13 Line

44 through Column 14 Line 7, i.e., *Replicators act as outside representatives of their*

regions to outside hosts and **Decisions on replica placement are done "in**

cooperation" between hosts and the replication service; Rabinovich Column 16

Lines 56-57, i.e., *Host s sends a replication request to E's replicator $r._E$, which includes*

the ID of the object to be replicated and the load on host s generated due to x_3. r_E

forwards this request along the path with the lowest-loaded hosts. When this request

*reaches a host, the host checks if its load is below the low watermark, **in which case it***

***copies the object from s** or increments the affinity of an existing replica of the object,*

*updates its upper-bound load estimate, and **sends acknowledgment** back **up the tree***

***to** r_E. Each replicator on the path updates its upper-bound estimates of the minimum*

host load for itself and the child replicator involved);

        "a module for transferring the file to the second computer" (Rabinovich Column

16 Lines 55-60, i.e., *Host s*);

        "a module for storing information indicative of whether the file has been

transferred to the second computer or the file exists in the first computer" (Rabinovich,

Column 8 Lines 7-11, i.e., *Each host that stores a replica of the requested object is*

ranked in decreasing distance metric in relation to the requester, step 401; Column 14

Lines 25-30, i.e., *Affinity is a compact way of representing multiple replicas of the same*

object on the same host. When the replica is first created, its affinity is initialized to 1;

when an object is migrated or replicated to a host that already has a replica of this

object, its affinity is incremented; and Column 14 Lines 44 through Column 15 Line 18,

i.e., pseudocode showing the storing of information indicative of whether a file has been

transferred); and

"a module for storing a path name for the second computer when the file has

been transferred to the second computer" (Rabinovich, Column 13 Lines 63-65, i.e.,

computing preference paths; Column 14, Lines 7-24, i.e., *an entity that frequently*

*appears in **preference paths** may be a good candidate for placing an object replica*

and Column 6 Lines 7-29; (Note that request distributor/replicators in an area are

always in communication with other requestor distributors/replicators and advertisement

packets are inherent in the method Rabinovich, which notifies placement of replicas in

particular network areas (Column 13 Line 44 through Column 14 Line). Therefore,

requestors would know exactly where the transferred file is located via

distributors/replicators);

"a module for allowing the third computer to access the file via the first computer"

(Rabinovich, Column 7 Lines 39-44, i.e., ***The request distributor selects a host*** *that*

stores a replica of the requested object to respond to the request *based upon the*

request metric and the distance metric of the host in relation to the request metric and

distance metrics of the other hosts that also store replicas of the requested object, step

204; Rabinovich, Column 9 Lines 9-23, i.e., *a method for identifying a second host to*

which it is beneficial to migrate a replica from a first host to the second host is shown in

Figure 8; Rabinovich, Column 13 Lines 63-65, i.e., ***computing preference paths;***

Column 14, Lines 7-24, i.e., *an entity that frequently appears in **preference paths** may*

be a good candidate for placing an object replica ; These disclosures of Rabinovich

teaches (1) Request Distributor could route requests from the third computer (the

Requestor) to a host which stores the replica or migrated replica by way of preference

paths;) "based on the information and the path name when the first computer receives

an access request for the file" (Rabinovich, Column 13 Lines 63-65, i.e., **computing**

preference paths; Column 14, Lines 7-24, i.e., *an entity that frequently appears in*

preference paths *may be a good candidate for placing an object replica*; Rabinovich,

Column 7 Lines 39-44, i.e., **The request distributor selects a host** *that* **stores a**

replica) "after the file has been transferred to the second computer" (Rabinovich,

Column 8 Line 65 through Column 9 Line 8, i.e., *a method for offloading*; Column 9

Lines 8-11, i.e., *a method for identifying a second host to which it is beneficial to migrate*

a replica from a first host to the second host is shown in Figure 8);

"a module for receiving a return request packet from the second computer" and

issue a read request in response, for returning the file to the first computer" (Rabinovich,

Column 17 Line 28 through Column 18 Line 40, i.e., *When host s is in* **Offloading**

mode, *it migrates or replicates objects to other nodes even if it is not beneficial*

geographically. The protocol contains two stages. In the first stage the protocol

identifies a recipient of the objects from s. The recipient is chosen to be the nearest

node to s whose load is below lw. This stage is shown in the following pseudo code

embodiment: Offload Request (s): /*Executed by the replicator r*/ let r_{min} be the

replicator with the smallest minimum host load among child replicators of r; if

min_load(r_{min}) < lw send Recipient request(s) to r_{min} ; if r_{min} responded with OffloadDone

(load) UpdateMinLoads(r_{min} load); exit; elseif $r._{min}$ responded with OffloadFailed (load)

UpdateMinLoads ($r._{min}$, load); send Offload Request(s) to parent unless r is the root;

endif else send Offload Request(s) to parent unless r is the root; end Recipient

request(s): /* executed by node q */ if q is a host if load(q) < lw initiate offloading from s by sending OffloadOffer (load(1)) to s; if any objects were offloaded send OffloadDone (load((q) to invoker; else send OffloadFailed (load(q))to invoker; endif else send Offload Filed (load (q)) to invoker; endif exit; endif let $r._{min}$ be the replicator with the smallest minimum host load among child replicators of q; if min_load ($r._{min}$) < lw send Recipient request(s) to $r._{min}$; if $r._{min}$ responded with Offload Done (load) UpdateMinLoads ($r._{min}$, load); send OffloadDone (min_load(q)) to invoker; elseif $r._{min}$ responded with OffloadFailed (load) Update MinLoads ($r._{min}$, load); send OffloadFailed (min_load (q)) to invoker; endif else Send OffloadFailed (min_load(q)) to invoker; endif end ;   This stage is initiated by the request for offloading from s to its parent replicator. This request travels up the tree until it find the replicator that has a child replicator with the minimum host load below lw. Then, this replicator send Recipient request message down the tree along the path with the lowest load until it reaches a host q. Upon receiving Recipient request, host q starts the second stage of the protocol by sending **OffloadOffer** to s. A pseudo code illustration of the method for this stage is shown as follows:

OffloadOffer (recipient_load): /*Executed by the offloading host s*/ while load(s) > lw AND recipient_load < lw AND not all objects have been examined let x.sub.3 be the unexamined object with the highest value of cnt(E,$x._s$) for some E if cnt($x._s$))/aff($x._s$)) < m send MigrateRequest($x._s$), load($x._s$))/aff($x._s$)) to q; load(s) = load(s) - load($x._s$))/aff($x._s$)) - 0.5 * load($x._s$))/(aff($x._s$)) - 1); recipient_load = recipient_load + 4 * load($x._s$))/aff(x.sub.s); decrement aff($x._s$)) if it was greater than 1, or drop $x._s$), otherwise; else send ReplicateRequest(x.sub.s,load($x._s$))/aff($x._s$)) to q; load(s) - load(s) - .66 *

*load(x.ₛ))/aff(x.ₛ)); recipient_load = recipient_load + 4 * load(x.ₛ))/aff(x.ₛ)); endif;*

endwhile; end : The offloading host goes through all its objects, starting with those that

have a higher rate of "foreign" requests, and attempts to send them to the recipient.

Unlike in the DecidePlacement method, **the offloading host** *does not try to migrate*

heavily loaded objects--objects with the relative access count above replication

threshold m can only be replicated. The reason is that load-migrating these objects

outright undo a previous replication. Particularly note that these disclosures and pseudo

codes of Rabinovich teach that the second computer is making a offloading request or

OffloadOffer (i.e., return request packet from the second computer) to other

computers (inherently including the first computer which is the source computer from

which most file(s) were initially transferred to the second computer).

However, Rabinovich does not explicitly recite that the offloading of files was

back to the first computer per se. Therefore, Rabinovich only inherently teaches the

limitation: "module for receiving and storing the file in the first computer, when the file is

returned from the second computer"".

On the other hand, Fun teaches said limitation:

"module for receiving and storing the file in the first computer, when the file is

returned from the second computer" (Fung, *Figure 1a, i.e.,* S1 TRS1 on 110 and S2 on

TRS2 on 120; Fung Paragraph 0045, i.e., *the migratable framework detects that the S1*

is down in step 230. In one embodiment, the user issued command in step 220 informs

the migratable framework that the server is down. When the migratable framework

detects the server is down in step 220, the migratable **framework moves the TRS to a**

back-up server. The back-up server may be specified by a user or be pre-determined

by the migratable framework system. After step 230, the migratable framework then

*activates **TRS1 on S2** in step 240;* Note that TSR1 was migrated to S2 and at this point

TSR1 is residing on server S2;   Fung Paragraph 0046, i.e., *Next, **TRS1 migration***

***back to S1** is initiated in step 420. In one embodiment, an administrator may manually*

*initiate **migration of the TRS back to the original server***. *In another embodiment,*

*migration is initiated when TRS1 contacts the migratable framework and **makes a***

***request to migrate TRS1 back to S1**. In step 430, the migratable framework*

completes the migration of TRS1 from S2 back to S1. Note herein that "a request to

migrate TRS1 back to S1" is equivalent to "the return request packet from the second

computer" of the claimed invention).

At the time the invention was made, it would have obvious to a person of ordinary

skill in the art to modify the computer system of Rabinovich to add the feature of issuing

a return request packet from the second computer to the first computer and storing the

returned file at the first computer, as taught by Fung, to the computer system of

Rabinovich so that the resultant method would comprise a module for receiving a return

request packet from the second computer and issue a read request in response, for

returning the file to the first computer and module for receiving and storing the file in the

first computer, when the file is returned from the second computer. One would have

been motivated to do so in order to provide a migration system which keeps all servers

virtually up during maintenance or crashes (Fung paragraph 0012) by migrating

modules on non-functioning servers to functioning servers and re-migrating said

modules back to their parent servers when parent servers are back up again (Fung

paragraph 0045-0046).

As per claim 2, Rabinovich in view of Fung is directed to the limitations:

"wherein: the memory stores a path of the file accessed by the third computer

associating the with information on the access situations of the third computer"

(Rabinovich Column 14, Lines 7-24, i.e., *total access count*) , and

"the program code further includes a module for designating the file

corresponding to the access situation information as the candidate for migration when

the information satisfies a predetermined condition" (Rabinovich Column 14, Lines 7-

24, i.e., *So, an entity that frequently appears in preference paths may be a good*

candidate for placing an object replica).

As per claim 4, Rabinovich in view of Fung teaches the limitation:

"wherein: the third computer comprises a memory for storing data and a

program code" (Rabinovich Figure 1: *Host 103, Host 104,Host 105, and Request*

Distributor 101, and Column 7 Lines 7-49), and

"the program code in the memory of the third computer includes a module for

receiving the advertisement packet and a module for making access to the second

computer for the file according to the advertisement packet" (Rabinovich Column 6

Lines 7-29, i.e., *and distribute the request to a host (e.g. host 103) that stores a replica*

of the requested object). Note that request distributor/replicators in an area are always in communication with other requestor distributors/replicators and advertisement packets are inherent in the method Rabinovich, which notifies placement of replicas in particular network areas (Rabinovich Column 13 Line 44 through Column 14 Line).

As per claim 5, Rabinovich in view of Fung teaches the limitation:

"wherein: the first network is further connected to a third network, and the program code further includes a module for transmitting the migrator acceptor search packet to the third network when no computer suitable for accepting the file is found in the second network" (Rabinovich Figure 1: *Host 103, Host 104,Host 105, Request Distributor 101, Network 102, and Requestor 109* and Column 16 Lines 23 through Column 17 Line 22). Note that the method of Rabinovich has replicators in a hierarchy (Rabinovich Column 13 Line 44 through Column 14 Line 25). If there is no computer suitable for accepting the file, acceptor search packets are inherently sent to other replicators in the higher levels of the hierarchy until a suitable one is found. Since the replicators are connected to Internet, there are more than one network, thus second, third, and more networks.

Claim 9 is essentially the same as claim 1 except that it set forth the claimed invention as a first computer which is connected to a first network capable of communicating with a second network including a second computer and a third computer and which ha a file accessed by the third computer rather than a computer

system comprising a first network, a first computer connected to the first network, a second network connected to the first network, and a second computer and a third computer connected to the second network.

Claim 12 is essentially the same as claim 1 except that it set forth the claimed invention as a program stored in memory of a first computer which is connected to a first network capable of communicating with a second network including a second computer and a third computer and which has a file accessed by the third computer rather than a computer system comprising a first network, a first computer connected to the first network, a second network connected to the first network, and a second computer and a third computer connected to the second network.

As per claim 14, Rabinovich in view of Fung teaches the limitation:

"wherein the program code further includes a module for transferring a directory belonging to the file to the second computer" (Rabinovich Column 16 Lines 55-60, i.e., *Host s*). Note that the system and methods of Rabinovich migrating a replica, which could be a file or files or a directory or directories.

As per claim 15, Rabinovich in view of Fung teaches the limitation:

"wherein the program code further includes a module for transmitting the path name when the first computer receives an access request for the file" (Rabinovich, column 10 Line 31 through Column 12 Line 16).

As per claim 16, Rabinovich in view of Fung teaches the limitation:

"wherein the file stored into the second computer when the file is transferred form

the first computer to the second computer" (Rabinovich, Column 7 Line 38 through

Column 8 Line 31, i.e., *The request distributor selects a host that stores a replica of the*

requested object to respond to the request based upon the request metric and the

distance metric; and *the request distribution decision as to which host to assign the*

request is made in accordance with the method shown in Figure 3).

As per claim 17, Rabinovich in view of Fung teaches the limitation:

"wherein the file is returned from the second computer to the first computer

depending on another access situation" (Rabinovich, Column 6 Lines 64-67, i.e., *The*

request metric for a replica is a historical measure of the request for the object that have

been forwarded to the host; Rabinovich, Column 15 Lines 38-46, i.e., *A host s can be in*

one of the two modes of operation. If its load exceeds high-water mark hw, it switches to

an offloading mode, where it sheds objects to other hosts, even if it is not geographically

beneficial. Once in this mode, the host continues in this manner until its load drops

below a low water mark, lw. Then, it moves objects only if it is geographically beneficial,

and stays in this mode until its load again exceeds hw. Water-marking is a standard

technique to add stability to the system; and Yamada, Paragraph 0095, i.e., *the system*

may be adapted so that the log file 10 is downloaded from the server 100 to and

provided in the client 200, that the log file 10 is updated in the client 200 according to

the execution state of the installation in the client 200, and that the log file 10 is returned

to the server 100 before the client 200 is rebooted, and downloaded to the client 200

after rebooted, and thereafter updated according to the execution of the installation in

the client 200).

As per claim 19, Rabinovich in view of Fung teaches the limitation:

"wherein the module for transferring the file to the second computer is performed

if the response indicates that the second computer accepts the file and the second

computer has a capacity for storing the file" (Rabinovich, Column 11 Line 30 through

Column 12 Line 17).

As per claim 20, Rabinovich in view of Fung teaches the limitation:

"wherein the first computer transfers the file to the second computer"

(Rabinovich, Column 8 Line 65 through Column 9 Line 8, i.e., *a method for offloading;*

Column 9 Lines 8-11, i.e., a method for identifying a second host to which it is beneficial

to migrate a replica from a first host to the second host is shown in Figure 8), "according

to the access request for the file **or** according to the reduction of an amount of the

access packets" (Rabinovich, Column 8 Lines 65 through Column 9 Lines, i.e., *a*

method for offloading in accordance with an embodiment of the present invention is

*shown in FIG 7. A second host identified **whose load is below a predetermined value***

lw (called the low water mark), such that lw is smaller than the hw, step 701), "after the

advertisement is sent" (Calo, Paragraph 0047).

Claim 21 is essentially the same as claim 1 except that it set forth the claimed invention as a method rather than a computer system comprising a first network, a first computer connected to the first network, a second network connected to the first network, and a second computer and a third computer connected to the second network.

Claim 22 is essentially the same as claim 14 except that it set forth the claimed invention as a method rather than a computer system comprising a first network, a first computer connected to the first network, a second network connected to the first network, and a second computer and a third computer connected to the second network.

Claim 23 is essentially the same as claim 15 except that it set forth the claimed invention as a method rather than a computer system comprising a first network, a first computer connected to the first network, a second network connected to the first network, and a second computer and a third computer connected to the second network.

Claim 24 is essentially the same as claim 16 except that it set forth the claimed invention as a method rather than a computer system comprising a first network, a first computer connected to the first network, a second network connected to the first

network, and a second computer and a third computer connected to the second

network.

9.   Claims18 is rejected under 35 U.S.C. 103(a) as being unpatentable over

Rabinovich in view of Fung and further in view of Sunna (U.S. Patent Application

Publication Number 2006/0036892).

As per claim 18, Rabinovich in view of Fung as applied to claim 17 teaches the

limitation: "the file is returned from the second computer to the first computer" (Fung

paragraph 0045-0046).

Rabinovich in view of Fung does not explicitly teach the limitation: "a module for

deleting the path name".

On the other hand, Sunna teaches the limitation:

"a module for deleting the path name" (Sunna, Paragraph 0071, i.e., *When an*

LSP is no longer needed in the network 100, ***an LSP delete*** *message is propagated*

through the nodes 110.sub.j defining the deleted LSP).

At the time the invention was made, it would have been obvious to a person of

ordinary skill in the art to add the feature of deleting a path, as taught by Sunna, to the

computer system of Rabinovich in view of Fung so that the resultant system would

comprise a module to delete a path name. One would have been motivated to do so in

order to reflect the new location of a migrated file, which is notoriously well known in the

art.

Contact Information

10.   Any inquiry concerning this communication or earlier communications from the
examiner should be directed to Dennis Myint whose telephone number is (571) 272-
5629. The examiner can normally be reached on 8:30AM-5:30PM Monday-Friday.

      If attempts to reach the examiner by telephone are unsuccessful, the examiner's
supervisor, John Breene can be reached on (571) 272-4107. The fax phone number for
the organization where this application or proceeding is assigned is 571-273-5629.

      Information regarding the status of an application may be obtained from the
Patent Application Information Retrieval (PAIR) system. Status information for
published applications may be obtained from either Private PAIR or Public PAIR.
Status information for unpublished applications is available through Private PAIR only.
For more information about the PAIR system, see http://pair-direct.uspto.gov. Should
you have questions on access to the Private PAIR system, contact the Electronic
Business Center (EBC) at 866-217-9197 (toll-free).

                                                            Dennis Myint
                                                            Examiner
                                                            AU-2162

/Cam Y Truong/

Primary Examiner, Art Unit 2162